

APLIKASI BIBLIOGRAFI PERPUSTAKAAN BERBASIS TEKNOLOGI WEB SEMANTIK

Mukhamad Nurkamid¹

ABSTRACT

Development of the technology the more days the more rapidly, almost all internet-based technology. One technology that is currently developing a world wide web, or known by the name of the web. The more information is added, the greater the size of the web and increasingly difficult to find information, because it needed a technology to get the information truly desired user. Writing the end of this task is one of the application of semantic web technology for the library by using the bibliographic languages RDF/OWL to create applications in the search system with search thesaurus.

Technology in the semantic web there are several approaches that can be used to implement thesaurus, one approach is SKOS (Simple Knowledge Organization System). SKOS effectively used as a semantics-based approach because it has several advantages, including providing more structure modeling for a vocabulary word (thesauri), classification and the list of terms (glossaries), which made possible a document can be linked with other documents, so that documents no longer stand on its own but a document that has a mutual relationship.

Some combination of semantic web technology over the results to provide users with information that is presented so that the wealthier one of several characteristics of the Semantic Web "intelligent applications" really happened.

Keywords: *web, semantic web, RDF/OWL, thesaurus, SKOS.*

ABSTRAKSI

Perkembangan teknologi kian hari kian pesat, hampir semua bidang teknologi berbasis internet. Salah satu teknologi yang saat ini terus berkembang adalah world wide web atau dikenal dengan nama web. Web kini menjadi sumber data yang terbesar dan sangat berharga, karena didalam web terdapat dokumen-dokumen saling terhubung dan banyak informasi yang dapat digali. Makin banyak informasi yang ditambahkan, makin besar ukuran web dan semakin sulit untuk mencari informasi, oleh karena itu dibutuhkan suatu teknologi untuk mendapatkan informasi yang benar-benar diinginkan pengguna. Pada penelitian ini merupakan salah satu penerapan teknologi semantic web untuk bibliografi perpustakaan dengan menggunakan bahasa RDF/OWL dalam membuat aplikasi pencarian dengan sistem pencarian thesaurus.

Didalam teknologi semantic web ada beberapa pendekatan yang dapat digunakan untuk mengimplementasikan thesaurus, salah satu pendekatan tersebut adalah SKOS (Simple Knowledge Organisation System). SKOS efektif digunakan sebagai salah satu pendekatan berbasis semantik karena memiliki beberapa kelebihan, diantaranya menyediakan banyak pemodelan struktur kosakata untuk padanan kata (thesauri), klasifikasi dan daftar istilah (glossaries), yang dimungkinkan sebuah dokumen dapat dihubungkan dengan dokumen yang

¹ Dosen Fakultas Teknik Universitas Muria Kudus

lainnya, sehingga dokumen tersebut tidak lagi berdiri sendiri melainkan sebuah dokumen yang saling memiliki relasi.

Dari beberapa perpaduan teknologi semantic web diatas mampu memberikan hasil kepada pengguna dengan informasi yang disajikan lebih kaya, sehingga salah satu dari beberapa karakteristik semantic web yaitu “intelligent applications” benar-benar terjadi.

Kata kunci: *web, semantic web, RDF/OWL, thesaurus, SKOS.*

I. PENDAHULUAN

a. Latar Belakang

Perkembangan teknologi saat ini kian hari kian pesat, hampir semua bidang teknologi sekarang berbasis internet. Salah satu teknologi yang saat ini kita rasakan manfaatnya adalah *world wide web* atau dikenal dengan nama *web*. *Web* kini menjadi sumber data yang terbesar dan sangat berharga untuk setiap pengguna karena didalam web kumpulan dokumen-dokumen saling terhubung dan dapat diakses melalui koneksi internet. Setiap individu didunia bebas menambahkan konten dalam *web*-nya. Makin banyak informasi yang ditambahkan, makin besar ukuran *web* dan semakin sulit pula untuk mencari informasi yang benar-benar yang diinginkan di *web*, oleh karena itu dibutuhkan suatu teknologi untuk mendapatkan informasi yang benar-benar diinginkan tersebut.

Hal tersebut juga dialami para pengguna informasi dalam memanfaatkan teknologi internet, salah satunya dalam *web digital library*. *Web-web digital library* kian hari semakin berkembang karena dipandang sebagai peluang potensial bagi dunia akademik untuk menyimpan koleksi yang ada dipergustakaan seperti buku, jurnal ilmiah, majalah, tugas akhir, skripsi dan tesis maupun disertasi. Teknologi yang tepat untuk mengembangkan, membangun dan mengelola koleksi dokumen perpustakaan yang benar-benar diinginkan pengguna merupakan tujuan kedepan semua pihak.

Tujuan perpustakaan, pengarsipan, yang ada di institusi adalah untuk memelihara dokumen yang memiliki harga keilmuan yang dapat diakses dimasa depan sama mudahnya diakses pada saat ini. Dengan semakin kecilnya media data, penyimpanan menggunakan media digital menjadi sangat menarik. Keuntungan penyimpanan menggunakan media digital adalah pencarian dapat dilakukan tidak hanya di antara masukan katalog elektronik, tetapi dapat menampilkan sepanjang keseluruhan isi dokumen. Permasalahan baru muncul disebabkan *floppy disk*, *hard disk* dan CD cepat

ketinggalan jaman. Perkembangan sistem komputer dan komponennya lebih cepat dari pada perkembangan media digital, sehingga menyebabkan masalah kompatibilitas.

Peneliti dan pengembang *web* mengusulkan tambahan informasi bagi konten web yang sering diistilahkan dengan metadata. Istilah metadata ini sering dipakai dalam *semantic web*, yang menggambarkan pendekatan dalam menangani dan menyimpan dokumen. Saat ini dokumen mulai disajikan dalam format XML (*Extensible Markup Language*). Bahasa lain yang digunakan dalam mendukung visi *Semantic Web* selain XML adalah RDF/OWL, Selanjutnya RDF/OWL dengan kemampuan dan fasilitas yang dimilikinya mampu digunakan untuk merepresentasikan makna istilah dalam *web* hingga dapat diproses dalam mesin.

Implementasi penerapan teknologi *semantic web* yang penulis lakukan yaitu pada perpustakaan dengan menampilkan daftar bibliografi, baik untuk majalah, buku, jurnal, dan tesis. Pengelolaan buku, jurnal, tesis yang berfungsi sebagai literatur yang disusun secara sistematis ini sering dikenal dengan istilah bibliografi (Neil, 2004). Ada berbagai macam pelayanan yang disediakan didalam perpustakaan salah satunya yaitu pencarian untuk koleksi dokumen yang merupakan studi kasus pada penelitian ini.

b. Permasalahan

Dari latar belakang yang dikemukakan diatas, maka yang menjadi rumusan masalah dalam penelitian ini adalah :

1. Bagaimana mengimplementasikan pemodelan ontologi bibliografi perpustakaan dengan menggunakan bahasa RDF/OWL untuk pencarian yang lebih kaya dan efektif.
2. Setelah ontologi dibangun kemudian bagaimana membuat aplikasi pencarian dengan sistem pencarian *thesaurus* dengan menggunakan pendekatan SKOS (*Simple Knowledge Organization System*) yang menjadi standar hubungan berbasis semantik (*semantic relationship*).
3. Bagaimana mengimplementasikan aplikasi pencarian menggunakan JSP (*Java Server Page*), Jena API dan SPARQL sebagai bahasa *query* RDF/OWL.

c. Ruang Lingkup Penelitian

1. Bibliografi yang ditampilkan meliputi judul (*title*), penulis (*author*), tahun (*year*), abstrak (*abstract*) dan daftar pustaka (*citations*).
2. Pemodelan ontologi dibangun dengan menggunakan tool *protege 3_4*.
3. Sistem yang dibangun hanya untuk fasilitas pencarian.

4. Sistem pencarian yang dibangun dilakukan berdasarkan *title, author, year, adviser, publisher* dan tidak berdasarkan *abstract* dokumen.

II. TINJAUAN PUSTAKA

Thesaurus berasal dari bahasa Yunani yang artinya “sebuah himpunan yang berharga” atau lebih spesifik lagi “sebuah kekayaan” (*thesaurry*). Menurut Pendit (2008) *thesaurus* didefinisikan sebagai klasifikasi untuk istilah-istilah yang saling berkait membentuk struktur bahasa sehingga sebuah kata dapat dipahami dengan kata lainnya. Baeza dan Riberio (1999) menambahkan sedikit konsep *thesaurus* dari Pendit yaitu sebuah daftar yang sudah dibuat sebelumnya, berisi kata-kata penting dalam satu bidang tertentu yang dihimpun dalam sebuah sistem dan untuk setiap kata didalam daftar tersebut, terdapat kaitan dengan kata-kata lainnya.

1. RDF/OWL

RDF (*Resource Description Framework*) merupakan bahasa yang digunakan untuk merepresentasikan metadata. RDF mendukung *interoperabilitas* antar aplikasi yang mempertukarkan informasi yang bersifat *machine-understandable* di *web*. *Semantic web* terdiri dari data yang ditulis dalam bahasa yang dapat dimengerti oleh mesin, seperti RDF. RDF menggunakan graf untuk merepresentasikan kumpulan pernyataan. Simpul dalam graf mewakili suatu entitas, dan tanda panah mewakili relasi antar entitas.

RDF didasarkan pada gagasan dimana hal-hal yang sedang diuraikan memiliki properti yang didalamnya mempunyai nilai-nilai dan *resource* yang dapat diuraikan dengan pembuatan statemen (Manola dan Miller, 2004). RDF menggunakan istilah tertentu untuk menguraikan suatu statemen. Secara rinci bagian yang mengidentifikasi didalam statemen dapat dikatakan sebagai *subject*, karakteristik (*property*) dari *subject* disebut sebagai *predicate*, sedangkan nilai dari *property* disebut sebagai *object*. Lasila dan Swick (1999) menyatakan model data RDF terdiri atas tiga obyek tipe:

a. Sumber daya (*Resource*)

Segala sesuatu yang digambarkan dengan RDF disebut *resource*. *Resource* ini bisa berupa keseluruhan halaman *web*, misalnya dokumen HTML <http://www.w3.org/Overview.html>, bisa juga berupa suatu bagian dari sebuah halaman *web*. *Resource* ini biasanya diberi nama menggunakan URI (*Uniform Resource Identifiers*). Segala sesuatu bisa memiliki sebuah URI, karena URI bersifat bisa diperluas maka URI bisa digunakan sebagai pengenalan bagi berbagai macam entitas.

b. Properti (*Property*)

Properti merupakan aspek, karakteristik, atribut, atau relasi khusus yang digunakan untuk menggambarkan sebuah *resource*. Setiap properti memiliki arti khusus, mendefinisikan nilai yang mungkin, tipe *resource* yang digambarkan, dan relasinya dengan properti lain.

c. Pernyataan (*Statement*)

Suatu *resource* tertentu bersama dengan properti dan nilai dari properti untuk *resource* tersebut membentuk suatu pernyataan RDF. Ketiga bagian ini disebut subjek, predikat, dan obyek, membentuk *RDF triple*. Obyek dapat berupa *resource* lain, atau berupa literal (string sederhana atau tipe data primitif lain yang didefinisikan oleh XML) .

Model data RDF menyediakan suatu kerangka konseptual untuk mendefinisikan dan menggunakan metadata. Untuk menciptakan dan mempertukarkan metadata dibutuhkan sintaks konkret. Manola dan Miller (2004) mengemukakan spesifikasi RDF ini biasanya menggunakan dasar sintaks XML karena itu disebut juga dengan RDF/XML. Berikut salah satu contoh penggunaan sintaks RDF yang direpresentasikan dalam RDF/XML.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#"
>
  <contact:Person
    rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

Gambar 1. Contoh Sintaks RDF (Manola dan Miller, 2004)

OWL (*Ontology Web Language*) bahasa ontologi yang direkomendasikan oleh W3C untuk merepresentasikan arti dari istilah-istilah suatu domain pengetahuan secara eksplisit serta relasi antara istilah tersebut yang tidak dapat dipisahkan dengan dokumen dan aplikasi *web*. OWL dikembangkan dari teknologi lain yang direkomendasikan oleh W3C, yaitu XML/S dan RDF/S.

OWL menambahkan kosakata untuk menggambarkan kelas dan properti, yaitu antara lain: relasi antar kelas misalnya *disjoint*, kardinalitas, *equality*, tipe properti yang lebih kaya, karakteristik properti misalnya simetri, dan kelas enumerasi. Dengan demikian OWL memiliki fasilitas lebih dibanding XML dan RDF/S dalam mengekspresikan arti

dan semantik. OWL juga melebihi bahasa-bahasa lain tersebut di atas dalam kemampuan merepresentasikan isi dokumen *web* yang *machine-interpreted* (Bechhofer, dkk, 2004).

Ontologi OWL merupakan graf RDF yang diubah menjadi kumpulan *triple* RDF. OWL menggunakan sintaks yang sama dengan RDF dalam merepresentasikan *triple*. Sintaks tersebut memiliki arti yang sama dengan representasi sintaks RDF/ XML berikut :

```
<rdf:Description rdf:about="#Continent">
<rdf:type
rdf:resource="http://www.w3.org/2002/07
/owl#Class"/>
</rdf:Description>
```

Gambar 2. Sintaks RDF/ XML yang sama arti dengan sintaks *instance* pada OWL (Bechhofer, dkk, 2004)

OWL merupakan perluasan dari kosakata RDF. Setiap graf RDF dapat membentuk ontologi OWL. Arti sebuah graf RDF sama dengan arti yang diberikan oleh graf OWL. Informasi dalam bentuk OWL disatukan dalam ontologi, yang dapat disimpan sebagai dokumen di *web*.

2. SKOS

SKOS (*Simple Knowledge Organisation System*) adalah desain dari skema RDF yang distandarkan oleh W3C yang digunakan untuk membantu merepresentasikan dan mengatur kosa kata didalam teknologi *semantic web* yang dapat diimplementasikan dengan bahasa ontologi (Miles, dkk, 2005). Sementara Miles dan Brickley (2005) menambahkan definisi sebelumnya bahwa misi dari SKOS ini menyediakan *framework* yang lengkap untuk mengekspresikan sistem organisasi pengetahuan melalui cara bagaimana data dapat dipahami oleh mesin (*machine-understandable*). Definisi lain ditambahkan oleh Miles dan Bechhofer (2009) menambahkan definisi sebelumnya bahwa SKOS adalah sebuah model data secara umum untuk berbagi dan menghubungkan sistem organisasi pengetahuan melalui *web*. Didalam penggunaan SKOS ini terdapat istilah-istilah (*term*) yang harus didefinisikan diawal.

Miles, dkk (2005) menyatakan untuk merepresentasikan beberapa jenis kosa kata dilingkungan RDF ini sudah banyak diterapkan, beberapa diantaranya untuk *glossary*, *taxonomy* dan *thesaurus*.

Isaac dan Summers (2009) menyimpulkan pada dasarnya SKOS mempunyai banyak karakteristik didalam pemodelan data, beberapa karakteristik yang dimiliki SKOS tersebut

diantaranya, konsep dari sumber daya (*conceptual resources*) yang dapat diidentifikasi oleh URI, pemberian nama (*labels*), hubungan semantik (*semantically relation*) agar informasi dapat terhubung dengan yang lainnya.

2.1. Konsep(*Concept*)

Unsur pokok dari kosakata SKOS disebut sebagai konsep (Isaac dan Summer, 2009). Konsep merupakan bagian dari pemikiran (*ide*), maksud/ arti atau kategori dari sebuah obyek dan kejadian-kejadian yang mana dijadikan dasar untuk mengorganisasikan sistem pengetahuan (*knowledge organization systems*). SKOS memberikan pengenalan terhadap properti ini dengan nama kelas `skos:Concept`, yang mana mengizinkan pengguna untuk menyatakan *resource* yang ingin didefinisikan. Hal ini biasanya dilakukan dengan dua langkah:

- Dengan membuat atau menggunakan sebuah URI (*Uniform Resources Identifier*)
- Dengan menyatakan didalam dokumen RDF, yang dapat menggunakan properti `rdf:type` bahwa *resource* dikenali oleh URI dari jenis `skos:Concept`.

2.2. Nama (*Labels*)

Nama menjadi bagian dari karakteristik sebuah konsep karena digunakan untuk mengekspresikan bahasa. SKOS menyediakan tiga istilah kekayaan untuk pemberian nama (*labels*) untuk konsep sebuah *resource*: `skos:prefLabel`, `skos:altLabel` dan `skos:hiddenLabel`.

2.2.1. Preferred Lexical Labels

Properti `skos:prefLabel`, merupakan properti yang menegaskan “*prioritas label*” terkait dengan kosakata yang didefinisikan didalam *resource*. Gambar 3 menggambarkan bagaimana suatu konsep dilabelkan dengan priopritas di dalam sebuah kelas “*animals*”:

```
ex:animals rdf:type skos:Concept;  
skos:prefLabel "animals"@en;  
skos:prefLabel "animaux"@fr.
```

Gambar 3. Pemberian label “*preferred Label*” pada sebuah konsep (Isaac dan Summers, 2009)

2.2.2. Alternative Lexical Labels

Properti `skos:altLabel`, merupakan properti yang menegaskan “*alternatif label*” terkait dengan kosakata yang didefinisikan didalam *resource*. Ini sangat membantu ketika label yang menegaskan kosakata diluar “*prioritas label* “. Gambar 2.12 (a) contoh pemberian

label “skos:altLabel” untuk kelas “*animals*” dan gambar 2.12 (b) contoh pemberian label “skos:altLabel” untuk kelas “*rocks*”

```
ex:animals rdf:type skos:Concept;  
  skos:prefLabel "animals"@en;  
  skos:altLabel "creatures"@en;  
  skos:prefLabel "animaux"@fr;  
  skos:altLabel "créatures"@fr.  
  
ex:rocks rdf:type skos:Concept;  
  skos:prefLabel "rocks"@en;  
  skos:altLabel "basalt"@en;  
  skos:altLabel "granite"@en;  
  skos:altLabel "slate"@en.
```

Gambar 4. Pemberian label “skos:altLabel” (a) Pemberian label “skos:altLabel” yang diimplementasikan untuk kelas “*animals*” (b) Pemberian label “skos:altLabel” yang diimplementasikan untuk kelas “*rocks*” (Isaac dan Summers, 2009)

2.3. Hubungan Semantik (*Semantic Relationship*)

Pada dasarnya hubungan semantik (*semantic relationship*) ini sedikit lebih rumit dari jenis yang lainnya untuk menjelaskan sebuah konsep. Arti dari suatu konsep digambarkan tidak hanya oleh *natural-language* kata-kata dalam label-nya tetapi juga harus dihubungkan kekonsep lain di dalam kosa kata. Solusi yang digunakan untuk menghubungkan dengan konsep lain didalam kosa kata seperti *thesauri*, adalah dengan SKOS. SKOS menyediakan tiga kekayaan standar untuk merepresentasikan hubungan semantik, diantaranya (Isaac dan Summers, 2009) :

- a. skos:broader dan skos:narrower, memungkinkan penyajian dalam bentuk hubungan secara hirarki, sebagai contoh hubungan sebuah gaya (*genre*) dan jenis spesifik gaya tersebut atau tergantung penafsiran seperti hubungan antara bagian yang utuh (*whole*) dan bagian-bagiannya (*part*)
- b. skos:related, memungkinkan penyajian dalam bentuk asosiatif (tidak memiliki hubungan secara hirarki), seperti hubungan antara jenis peristiwa dan bagian dari peristiwa tersebut yang secara tidak langsung masih memiliki keterkaitan.

2.3.1. Broader/ Narrower Relationships

Untuk menyatakan sebuah konsep dalam arti yang lebih luas atau secara umum dari konsep yang lain dapat digunakan properti skos:broader. Sementara skos:narrower sebaliknya, yaitu untuk menyatakan sebuah konsep dalam arti sempit atau lebih spesifik dari konsep lainnya. Gambar 2.13 menunjukkan contoh kolaborasi kedua properti skos:broader dan skos:narrower pada kelas “*animals*”


```

ex:animals rdf:type skos:Concept;
  skos:prefLabel "animals"@en;
  skos:narrower ex:mammals.
ex:mammals rdf:type skos:Concept;
  skos:prefLabel "mammals"@en;
  skos:broader ex:animals.

```

Gambar 5. Properti skos:broader dan skos:narrower yang diimplementasikan pada kelas “animals” (Isaac dan Summers, 2009)

2.3.2. Associative Relationship

Hubungan asosiasi (*associative relationship*) digunakan untuk menyatakan hubungan diantara dua buah konsep. Contoh implementasi hubungan asosiasi antara kelas “bird” dan kelas “ornithology” dapat ditunjukkan pada gambar 6.

```

ex:birds rdf:type skos:Concept;
  skos:prefLabel "birds"@en;
  skos:related ex:ornithology.
ex:ornithology rdf:type skos:Concept;
  skos:prefLabel "ornithology"@en.

```

Gambar 6. Implementasi hubungan asosiasi antara kelas “birds” dan kelas “ornithology” (Isaac dan Summers, 2009)

3. SPARQL

SPARQL merupakan bahasa *query* untuk RDF/OWL. Di dalam lembar rekomendasinya Prud’hommeaux dan Seaborne (2005), W3C menuliskan SPARQL menyediakan fasilitas untuk mengekstrak informasi dalam bentuk URI, *blank node* dan literal, mengekstrak subgraf RDF, dan membangun graf RDF baru berdasar pada informasi dari graf yang di-*query*. *Query* SPARQL didasarkan pada pencocokan pola graf (Karsanti, 2006). Pola graf yang paling sederhana adalah *triple pattern* yang mirip dengan RDF *triple*, hanya saja pola pada *query* dimungkinkan pemberian nama diluar terminologi RDF pada posisi subyek, predikat dan obyek. Klausa yang digunakan dalam query SPARQL, diantaranya:

1. PREFIX

Statemen PREFIX merupakan sebuah metode yang digunakan sebagai penunjuk yang membawa informasi dalam suatu halaman *web*. Pada dasarnya PREFIX digunakan untuk menyingkat sebuah *resource*, dalam hal ini dapat diwakili oleh URI(*Uniform Resource Identifier*).

2. SELECT

Statemen SELECT didefinisikan sebuah daftar variabel-variabel yang akan dikembalikan sebagai hasil dari eksekusi *query*. Setiap variabel diawali dengan notasi (?).

3. WHERE

Statemen WHERE didefinisikan sederetan *triple pattern* yang harus dimiliki oleh setiap hasil *query* yang valid. Seluruh pola yang merepresentasikan suatu kalimat RDF harus sesuai dengan RDF *triples*, yaitu terdiri dari subyek, predikat dan obyek. Ketiga RDF *triple* tersebut dapat direpresentasikan oleh URI atau sebuah variabel dan literal.

4. OPTIONAL

Statemen OPTIONAL digunakan untuk mengatasi ketidakcocokan struktur pola *query* dengan pola yang ada pada graf RDF.

Teknologi *semantic web* menjadi fenomena yang cukup menarik untuk diteliti karena perkembangan teknologi *semantic web* dapat diterapkan di segala macam bidang, misalnya seperti di dunia pendidikan, *jurnalisme*, transportasi dan *inventory*.

Penelitian mengenai *semantic web* beserta standar pendukungnya telah banyak dilakukan. *Semantic web* merupakan perluasan dari *web* yang ada pada saat ini, di mana informasi memiliki arti yang terdefinisi lebih baik, sehingga memungkinkan manusia dan komputer untuk bekerja sama lebih optimal, terutama dalam pengelolaan dan penyajian informasi (Lee, dkk, 2001)

Penelitian tentang standar yang berfokus pada *content* untuk mendukung *semantic web* telah banyak dilakukan. Beberapa penelitian–penelitian mahasiswa ilmu komputer Universitas Gadjah Mada sebelumnya yang dijadikan referensi adalah “Model Ontologi untuk Jadwal Penerbangan menggunakan Protege”, (Minurita, 2005), “Pengembangan Model Ontologi untuk manajemen pengetahuan Staff Perusahaan”, (Riyanti, 2006), “Penerapan Ontology Web Language pada pengarsipan berita Online”, (Soraya, 2006) dan “Penerapan Teknologi *Semantic Web* pada Sistem Manajemen Perpustakaan Digital”, (Karsanti, 2006).

Persamaan dari penelitian sebelumnya adalah menggunakan teknologi *semantic web* untuk membangun sistem aplikasi pencarian sedangkan perbedaannya hanya terletak pada subyek (*domain*) dan metode sistem pencariannya saja, yang dalam hal ini penulis menggunakan pendekatan *thesaurus* dari penelitian yang sebelumnya yang masih menggunakan sistem berbasis pola. Studi kasus penulis untuk penelitian ini adalah bibliografi perpustakaan yang meliputi buku, majalah, jurnal, prosiding, skripsi dan tesis.

III. TUJUAN PENELITIAN

Tujuan penelitian ini adalah membuat aplikasi pencarian untuk bibliografi perpustakaan dengan menggunakan bahasa RDF/OWL sebagai bagian dari penerapan teknologi *semantic web*.

IV. METODA PENELITIAN

Metoda penelitian ini adalah menggunakan penerapan teknologi web semantik yaitu SKOS (*Simple Knowledge Organization Systems*) yang menjadi standar hubungan berbasis semantik (*semantic relationship*) yang telah direkomendasikan oleh W3C (*World Wide Web Consortium*).

V. HASIL DAN PEMBAHASAN

1. Perancangan Sistem

Perancangan sistem terbagi menjadi dua, yaitu perancangan ontologi untuk koleksi perpustakaan dan perancangan aplikasi. Perancangan ontologi menjelaskan langkah demi langkah pengembangan ontologi mulai dari penentuan domain, istilah/ terminologi, definisi kelas dan hirarki kelas, definisi properti, definisi konstrain dan pembuatan *instance*. Sedangkan perancangan aplikasi berbasis *semantic web* dijelaskan tentang gambaran umum aplikasi, analisis system dan spesifikasi kebutuhan sistem untuk aplikasi.

2. Perancangan Ontologi

Noy dan McGuinness (2000) telah menjelaskan ada beberapa langkah-langkah yang harus diperhatikan didalam pengembangan ontologi, salah satunya dengan menentukan konsep dan domain.

2.1. Penentuan Konsep dan Domain

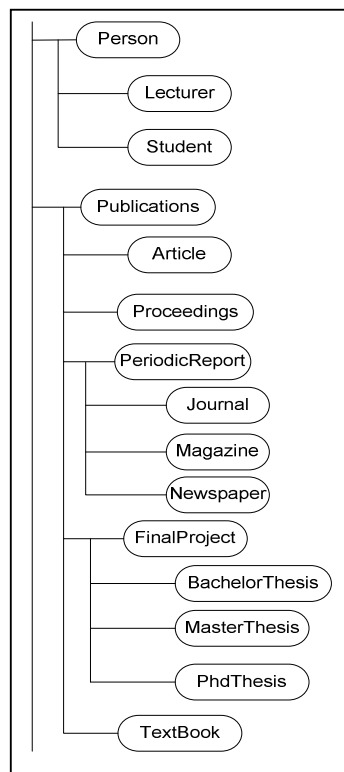
Penentuan konsep dan domain pada dasarnya merepresentasikan koleksi semua dokumen yang dilengkapi dengan informasi dan disusun berdasar suatu klasifikasi dan dikelompokkan kedalam jenis-jenis yang sama (*Class*). Dokumen tersebut meliputi *article*, *proceedings*, *textbook*, *periodicreport* (*journal*, *magazine*, *newspaper*) dan *finalproject* (*bachelorthesis*, *mastersthesis*, *phdthesis*).

2.2. Penentuan Daftar Terminologi

Penentuan daftar terminologi menegaskan hal-hal yang berkaitan dengan istilah-istilah yang digunakan didalam membuat statemen sekaligus memberikan jawaban dari statemen yang dibuat sebelumnya.

2.2.1. Definisi Kelas dan Struktur Hirarki

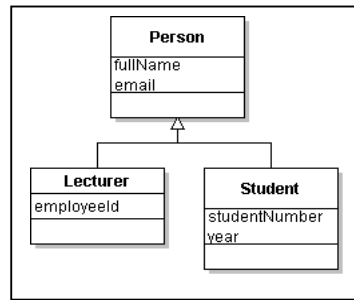
Representasi definisi kelas dan hirarki kelas adalah mengelompokkan kelas-kelas dengan karakteristik yang sama yang muncul didalam sebuah domain. Uschold dan Gruninger (1996) menegaskan didalam makalah “ *Ontologies: Principles, Methods and Applications*” bahwa ada beberapa pendekatan metode yang dapat digunakan untuk membangun struktur hirarki kelas diantaranya yaitu metode *top-down*, *bottom-up*, *combination*. Untuk aplikasi bibliografi ini digunakan pendekatan *top-down*, dimana kelas-kelas didefinisikan dari mulai konsep yang paling umum sampai konsep yang lebih spesifik.



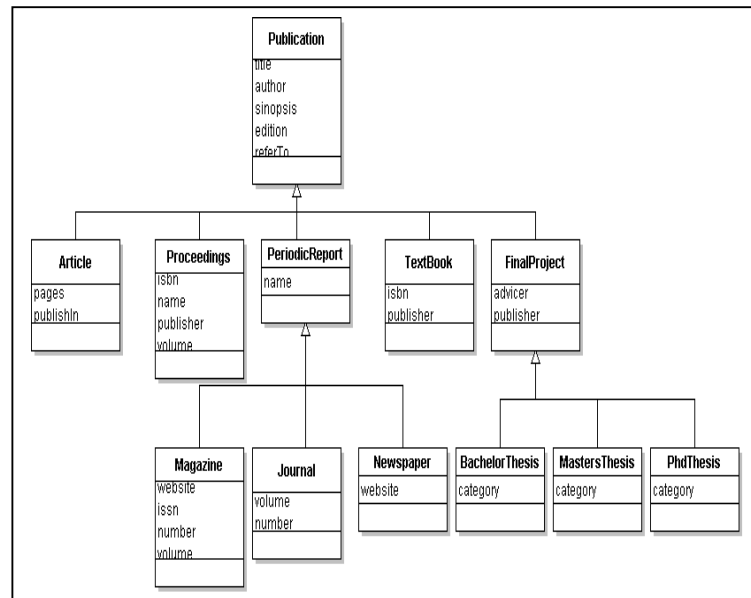
Gambar 7. Rancangan Kelas dan Struktur Hirarki Kelas Bibliografi

2.2.2. Diagram Kelas

Diagram kelas atau *class diagram* menunjukkan interaksi(relasi) antar kelas didalam sistem, sehingga dengan diagram kelas penyajian informasi yang dimiliki oleh setiap kelas dapat terlihat jelas. Diagram kelas pada aplikasi bibliografi dapat ditunjukkan dengan gambar 8.



(a)



Gambar 8. Diagram Kelas Bibliografi (a) Diagram Kelas Person (b) Diagram Kelas Publications

2.2.3. Definisi Properti (Slot)

Setelah kelas diciptakan langkah selanjutnya adalah mendefinisikan properti kelas. Sebuah kelas jika berdiri sendiri tidak akan memberikan informasi yang cukup tanpa adanya properti yang melekat didalamnya. Dengan properti inilah sebuah kelas akan mempunyai nilai tambah dalam hal ini adalah informasi.

2.2.4. Konstrain Properti

Konstrain properti merupakan batasan tertentu dimana properti yang dimiliki setiap kelas memiliki tipe nilai khusus. Didalam pengembangan ontologi konstrain properti dikategorikan menjadi dua kategori.

1. *Slot Kardinalitas*, *Slot* kardinalitas didefinisikan sebagai nilai banyaknya yang dimiliki setiap properti kelas.
2. *Slot Tipe*, *Slot* tipe menegaskan beberapa tipe data properti yang harus didefinisikan.

2.2.5. Pembuatan *Instance*

Langkah terakhir setelah konsep pembuatan properti kelas selanjutnya menciptakan sebuah *instance* dari kelas. Langkah pendefinisian sebuah *instance* kelas dimulai dengan memilih kelas, membuat individu *instance* dari kelas kemudian yang terakhir mengisi *slot* properti dari individu *instance* kelas.

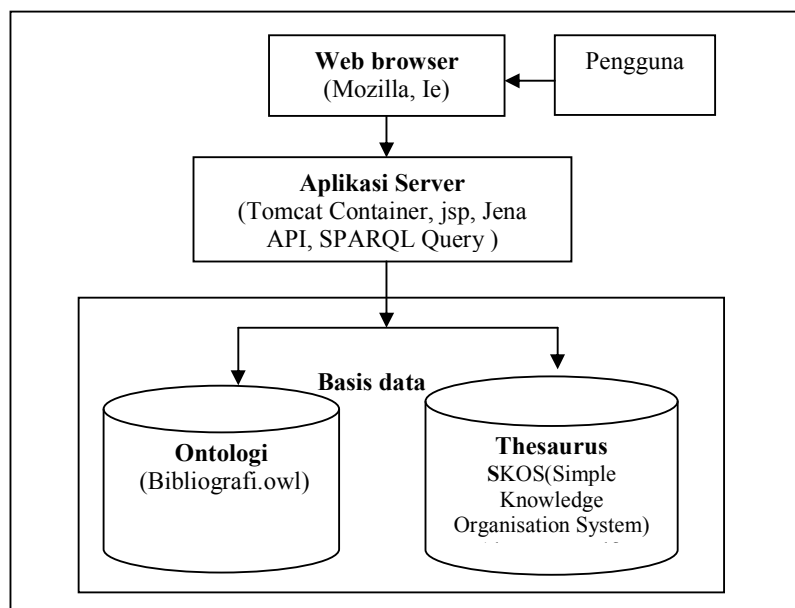
2.3. Perancangan Aplikasi

Rancangan aplikasi adalah sebuah sketsa dari aplikasi dimana sebuah aplikasi tersebut dimodelkan.

2.3.1. Gambaran Umum Aplikasi

Aplikasi yang dibangun dalam penelitian ini adalah aplikasi pencarian untuk koleksi perpustakaan yang merupakan salah satu bagian dari layanan sistem informasi dipergustakaan dengan menggunakan pendekatan *thesaurus*.

Perancangan aplikasi pencarian berbasis teknologi *semantic web* ini menggunakan pendekatan ontologi, dimana didalam ontologi tersebut digunakan sebagai penyimpanan dokumen bibliografi. Selanjutnya untuk menemukan informasi yang terdapat didalam ontologi dapat digunakan SPARQL sebagai bahasa *query* RDF/OWL, dan Jena API berfungsi sebagai *framework* yang menghubungkan antara ontologi dengan bahasa Java. Detail gambaran umum rancangan aplikasi dapat ditunjukkan oleh gambar 9.



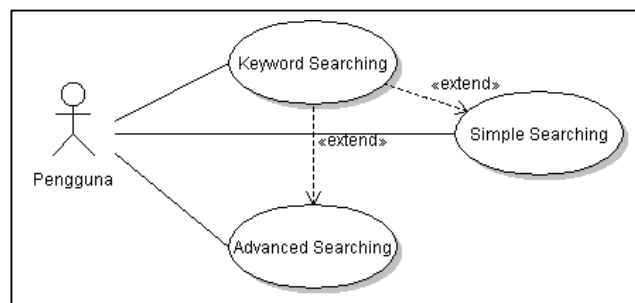
Gambar 9. Gambaran Umum Aplikasi Pencarian berbasis Teknologi *Semantic Web*

2.3.2. Analisis Sistem

Analisis sistem menegaskan penguraian dari suatu sistem informasi yang utuh ke dalam bagian komponen-komponennya.

2.3.2.1. Diagram Usecase Bibliografi

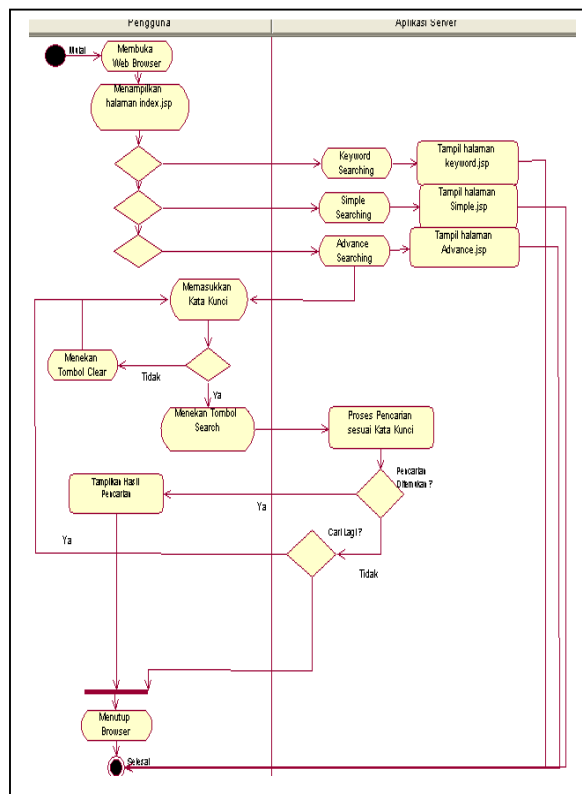
Diagram usecase bibliografi ini menunjukkan aktifitas yang dapat dilakukan oleh sistem. Dari gambar 10 dapat dijelaskan bahwa pengguna dimungkinkan dapat memilih beberapa menu pencarian yang ada, seperti *keyword searching*, *simple searching* dan *advance serching*. Notasi “*extend*” memberikan alternatif apabila dengan *keyword searching* tidak memberikan hasil sehingga pengguna dapat memilih menu yang lain.



Gambar 10. Diagram Usecase Pencarian Bibliografi

2.3.2.2. Diagram Aktifitas Pencarian Bibliografi

Aplikasi pencarian yang dibangun mempunyai tiga model pencarian, yaitu menggunakan *keyword searching*, *simple searching* dan *advanced searching*. Masing-masing model pencarian lebih jelas direpresentasikan dengan *activity diagram* yang diperlihatkan pada gambar 11.



Gambar 11. Diagram Aktifitas Pencarian Bibliografi

3. Implementasi

Setelah rancangan diciptakan, langkah selanjutnya adalah implementasi sistem. Pada segmen implemantasi penulis mencoba melakukan pendekatan pencarian berbasis pola dan pencarian berbasis *thesaurus*. Pencarian berbasis pola pada dasarnya pencarian berdasar *string*, dengan kata lain input yang diberikan dicocokkan dengan kata (*string*) yang sama yang ada didatabase, Implementasi aplikasi terdiri dari tiga menu utama pencarian, diantaranya *keyword searching*, *simple searching* dan *advance searching*. Keyword searching adalah fasilitas pencarian dimana pengguna dapat memasukkan beberapa kata kunci (*keyword*) pada judul bibliografi yang ada. Gambar 12 menunjukkan implementasi berbasis pola (*string*) pada menu keyword searching (a) desain tampilan (*interface*) dan (b) hasil implementasi dengan pencarian keyword searching.

(a)

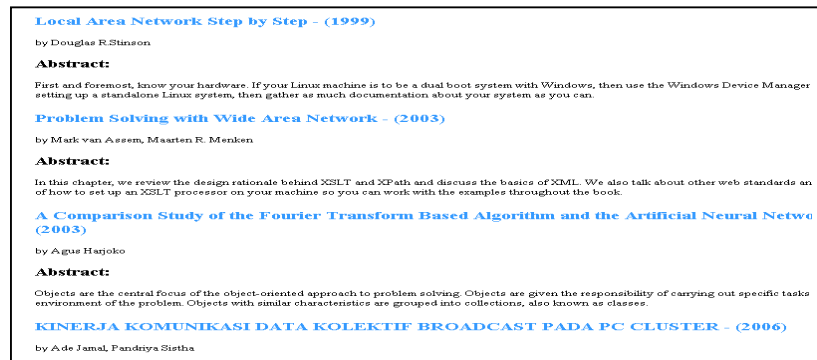
(b)

Gambar 12. Implementasi berbasis pola (string) pada menu *Keyword Searching* (a)

Desain interface keyword searching (b) Hasil pencarian dengan keyword searching

Pencarian berbasis *thesaurus* adalah pencarian yang berdasar pada klasifikasi untuk istilah-istilah yang saling berkait membentuk struktur bahasa sehingga sebuah kata dapat dipahami dengan kata lainnya (Pendit, 2008). Misalnya dalam hal ini istilah '*jaringan*' dapat dipahami dengan kata lainnya seperti '*network*', '*komunikasi data*', '*data communication*', '*neighborhood*' dan '*Local Area Network*'. Sehingga dari uraian diatas dapat disimpulkan bahwa sebuah istilah yang berkait tersebut pada prinsipnya tidak didasarkan pada struktur kata (*string*) yang sama melainkan berdasarkan pada makna (semantik). Implementasi pendektan pencarian berbasis thesaurus dapat ditunjukkan oleh desain interface menu simple searching pada gambar 13.

(a)



(b)

Gambar 13. Implementasi berbasis *thesaurus* pada menu *Simple Searching*
(a) Desain *interface* menu *Simple Searching* (b) Hasil pencarian dengan pendekatan *thesaurus*

VI. KESIMPULAN

Dari hasil implementasi yang diujicobakan dalam penelitian ini adalah:

1. Pendekatan ontologi adalah salah satu alternatif yang layak dipertimbangkan untuk penyimpanan basis data.
2. Dengan menggunakan SKOS (*Simple Knowledge Organisation System*) sebagai pilar manajemen kosakata, membuktikan mesin dalam hal ini komputer mampu mengetahui relasi antar dokumen-dokumen yang terhubung (*machine understandable*) sehingga dokumen tidak lagi berdiri sendiri.
3. Hasil implementasi dengan pendekatan SKOS untuk pencarian koleksi perpustakaan mampu memberikan jawaban yang selama ini pengguna keluhkan, yaitu informasi yang ditampilkan lebih kaya dan efektif.
4. Pada dasarnya ide *semantic web* bukanlah hanya berbicara mengenai pertukaran data, akan tetapi perlu di lihat bagaimana data didistribusikan dan disatukan, SKOS adalah salah satu mekanisme data dapat didistribusikan yang didukung bahasa RDF sebagai suatu format bahasa untuk pendistribusian data dilingkungan *web*.
5. Dengan pendekatan SKOS, paradigma pencarian yang semula berbasis teks (*text-based search*) dapat digantikan oleh pencarian berbasis konsep (*concept-based search*).

VII. DAFTAR PUSTAKA

- Baeza Y. R. dan Riberio N. B., 1999, *Modern Information Retrival*, ACM Press, New York.
- Isaac, A. dan Summers, E., 2009, *SKOS Simple Knowledge Organization System Primer*, Recommendation, World Wide Wibe Consortium (W3C) : <http://www.w3.org/TR/2009/WD-skos-primer-20090615/> available: <http://www.w3.org/TR/skos-primer>
- Karsanti, H. T., 2006, *Penerapan Teknologi Semantic Web pada Sistem Menejemen Perpustakaan Digital*, Skripsi, Program Studi Ilmu Komputer FMIPA UGM, Yogyakarta.
- Miles, A., Matthews, B., Beckett. D., Brickley, D., Wilson, M dan Rogers, N., 2005, *SKOS : A Language to describe simple knowledge structures for the web*, Recommendation, World Wide Web Consortium (W3C) : <http://www.w3.org/TR/swbp-skos:a language to describe simple knowledge structures for the web>.
- Miles, A. dan Bechhofer, S., 2009, *SKOS Simple Knowledge Organization System Reference* , Recommendation, World Wide Web Consortium (W3C):<http://www.w3.org/TR/2009/PR-skos-reference-20090615/>, Lastest version available : <http://www.w3.org/TR/skos-reference>
- Manola, F. dan Miller, E., 2004, *RDF Primer*, Recommendation, World Wide Web Consortium(W3C) : <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> Lastest version available : <http://www.w3.org/TR/rdf-primer/>
- Miles, A. dan Brickley, D., 2005, *SKOS Core Guide*, Recommendation, World Wide Web Consortium (W3C) : <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102>, available: <http://www.w3.org/TR/swbp-skos-core-guide>
- Miles, A., 2005, *Quick Guide to Publishing a Thesaurus on the Semantic Web*, Recommendation, World Wide Web Consortium (W3C) : <http://www.w3.org/TR/2005/WD-swbp-thesaurus-pubguide-20050517> Lastest version available: <http://www.w3.org/TR/swbp-thesaurus-pubguide>
- Miles, A., Matthews, B., Beckett. D., Brickley, D., Wilson, M dan Rogers, N., 2005, *SKOS : A Language to describe simple knowledge structures for the web*, Recommendation, World Wide Web Consortium (W3C) : <http://www.w3.org/TR/swbp-skos:a language to describe simple knowledge structures for the web>.

- Noy, N. F. dan McGuinness, D. L., 2000, *Ontology Development 101: A Guide to Creating Your First*, Stanford University, Stanford. Available http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html diakses tanggal 10 Januari 2009
- Pendit, P. L., 2008, *Perpustakaan Digital dari A-Z*, Cita Karyaarsa Mandiri, Jakarta.
- Seaborne, A. dan Prud'hommeaux, E., 2008, *SPARQL Query Language for RDF*, Recommendation, World Wide Web Consortium (W3C) : <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/> Lastest version available: <http://www.w3.org/TR/rdf-sparql-query/>
- Unschold, M. dan Gruninger, M., 1996, *Ontologies: Principles, Methods and Applications*, Knowledge Engineering Review 11(2). available: <http://www.aiai.ed.ac.uk/~oplan/documents/1996/96-ker-intro-ontologies.pdf> diakses tanggal 3 Maret 2009

DAFTAR RIWAYAT HIDUP



I. Data Pribadi

Nama : Mukhamad Nurkamid, S.Kom, M.Cs
Kelahiran : Pati, 20 Juni 1983
Status : Nikah
Alamat : Jatimulyo 03/ 01, Wedarijaksa, Pati 59152
Telpon : 081 225 104 30
E-mail : nurkamid@gmail.com
Agama : Islam

II. Pendidikan

- Tahun 2007- 2009 : Pasca Sarjana (S2) Ilmu Komputer UGM Yogyakarta
Tahun 2001-2005 : Sarjana (S1) Teknik Informatika UNISBANK Semarang
Tahun 2001-1998 : SMK Negeri 2 Pati - Jawa Tengah.
Tahun 1998-1995 : SLTP Negeri 1 Wedarijaksa, Pati – Jawa Tengah.
Tahun 1989-1995 : SD Negeri Jatimulyo, wedarijaksa, Pati – Jawa Tengah.

III. Pengalaman Bekerja

- 1) 2005-2007 : Teknisi dan Administrator Jaringan Komputer di SMK Negeri 2 Pati
- 2) 2005-2007 : Guru bid Studi Ketrampilan Komputer dan Pengelolaan Informasi (KKPI) di SMK Negeri 2 Pati
- 3) 2009-Sekarang : Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muria Kudus

IV. Seminar Nasional

1. Managing Internet Scurity Learn from Best Practice -2008
2. VOIP (Voice over Internet Protokol) -2008
3. Network-based Content to Optimize the Use of Information Technology in Business - 2008
4. Perkembangan Virus dan Anti Virus - 2008
5. Networking Security and Hacking - 2008
6. Menyambut Berlakunya Undang-undang Informasi dan Transaksi Elektronik (UU ITE) - 2008
7. Hardy Release Party and Talk Show (Linux Ubuntu 8.04) - 2008
8. IPv4/IPv6 Transition - 2008
9. Next Generation Networking (NGN) - 2008
10. JAMU- (Java Meet User group) - 2008
11. Free Open Source Software (FOSS) - 2009
12. The Power of Client-Side Scripting Technology AJAX (Asynchronous Javascript and XML) – 2009

V. Pelatihan/ Workshop

1. Object Oriented with Java - 2008
2. Web Programming with PHP - 2008
3. IPv4/ IPv6 Transition - 2008
4. Integrated Research Training (Interest) – 2008
5. Pelatihan Teknologi Informasi pengelola BDS (Pendukung FPESD) se- Jawa Tengah - 2008
6. Pelatihan TOEFL- 2008
7. Workshop AJAX (Asynchronous Javascript and XML) - 2009

IX. Keterangan Lain

Konsentrasi Komputer:

- Database (Mysql, SQLite 3.0, Database Firebird 2.0, SQL Server, Database XML (XQuery, Xpath, XSLT)
- OODB/db4o (SODA Object Query API, Native Query, QBE Query), Eclipse, Database Perst
- Database Protege 3_3.1 : OWL/RDF (SPARQL-Query)
- Pemrograman Web (HTML,XML, PHP, JSP)